Imperative and Declarative Languages

Low and High Revisited

The **imperative** form of a language is related to the giving of commands (or asking questions).

A **declarative** form of language relates to describing states or conditions that need to be met. The specifics of how the conditions should be met are unspecified.

```
To exit a room:
1. Walk to the door.
2. Stop.
3. Is the light on?
3b. If yes: Turn off the light.
4. Open the door.
5. Walk outside.
```

```
Saving Electricity Policy:

When a room is unoccupied, the
light should be switched off.
```

Pseudo code to switch off a light written in imperative and declarative styles

Imperative languages describe processes as how-to steps to be performed one after the other.

Imperative languages are closely related to the operation of the machine, as at the lowest level, the computer is told explicitly in machine language instruction, *exactly* what to do at every moment. As a result, **lower level** programming languages tend to be **imperative by nature**.

Many languages tend to mix aspects of both styles of language. **SQL**, while conceptually declarative, includes imperative constructs. **Python**, inherantly a step-by-step procedural language, also provides many forms to support **functional programming**, a declarative style of coding. (Python **list comprehensions** are a clear example of functional programming, the special syntax in large part derived from Haskell, a highly declarative language.)

Declarative languages describe conditions, constraints, or rules for which there may be many or even no solutions; the process necessary to obtain a solution is unspecified.

Declarative languages are more related to abtract thinking and logic. Programs (or documents) written using a declarative language can often be thought of as puzzles to be worked out or solved. As only a "minimal set of conditions" for satisfaction are specified, a given program may have more than one solution, or be "open" to taking other context-dependent constraints into account in a solution.

Declarative languages are by nature of a "higher level", removed from the actual operation of a processor (or any step-by-step process).

Declarative languages like LISP and PROLOG are closely related to the development of **Artificial Intelligence** research.

C++

PERL

C

Java

PHP

Python

Photoshop Macros

JavaScript

AppleScript

LISP

SQL

Haskell

Prolog

HTML

CSS

iTunes "Smart" Playlists

Regular Expressions